Exemplar-based Robust Coherent Biclustering

Kewei Tu
Iowa State University

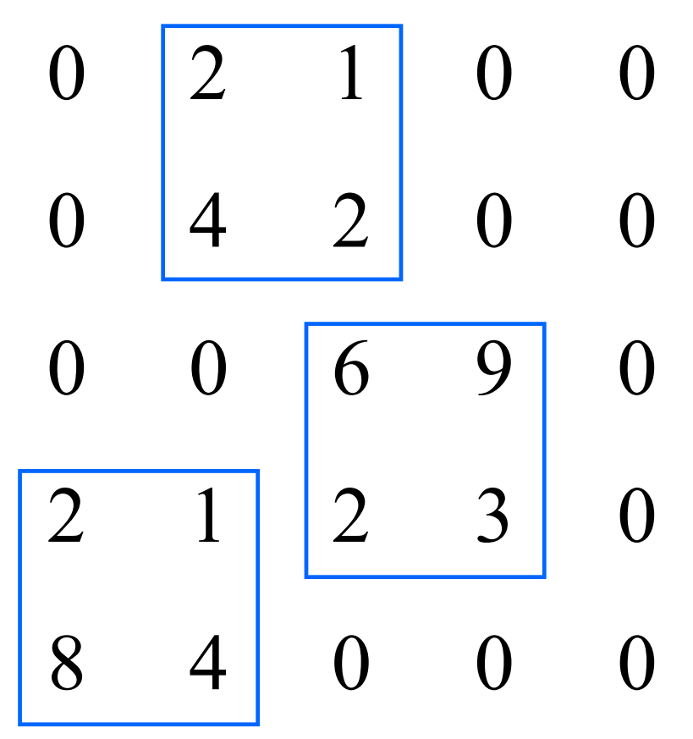
Xixiu Ouyang
Shanghai Jiaotong University

Dingyi Han
Shanghai Jiaotong University

Yong Yu
Shanghai Jiaotong University

Vasant Honavar
Iowa State University

1. Introduction



$$\begin{matrix} 0 & 2 & 1 & 0 & 0 \\ 0 & 4 & 2 & 0 & 0 \\ 0 & 0 & 6 & 9 & 0 \\ 2 & 1 & 2 & 3 & 0 \\ 8 & 4 & 0 & 0 & 0 \end{matrix}$$

Biclustering: simultaneously grouping the rows and columns of a data matrix to uncover sub-matrices (biclusters) that optimize a desired objective function; its applications include gene expression data analysis, document clustering, grammar rule induction, etc.

We introduce a novel formulation of coherent biclustering and use it to derive two algorithms. Our algorithms offer desirable features like finding arbitrarily positioned and possibly overlapping biclusters, being robust in the presence of noise and missing elements, automatically determining the number of biclusters, etc. In addition, our algorithms have two distinct features:

- Identification of an **exemplar element** (i.e., a representative row-column pair) of each bicluster, which assists in the interpretation of the resulting biclusters and increases the robustness of the algorithms in the presence of outliers.
- Being **robust** in the presence of interference from background elements (especially in the case that a row or column of background elements can fit into a coherent bicluster according to the coherence measure).

2. Formulation

The objective function is the sum of three components.

Bicluster coherence

$$K(B)=\sum_{b\in B}\sum_{\substack{i\in b.\text{rows}\\ j\in b.\text{columns}}} w_{ij}\, d(a_{ij},a_{ij}^{*})$$

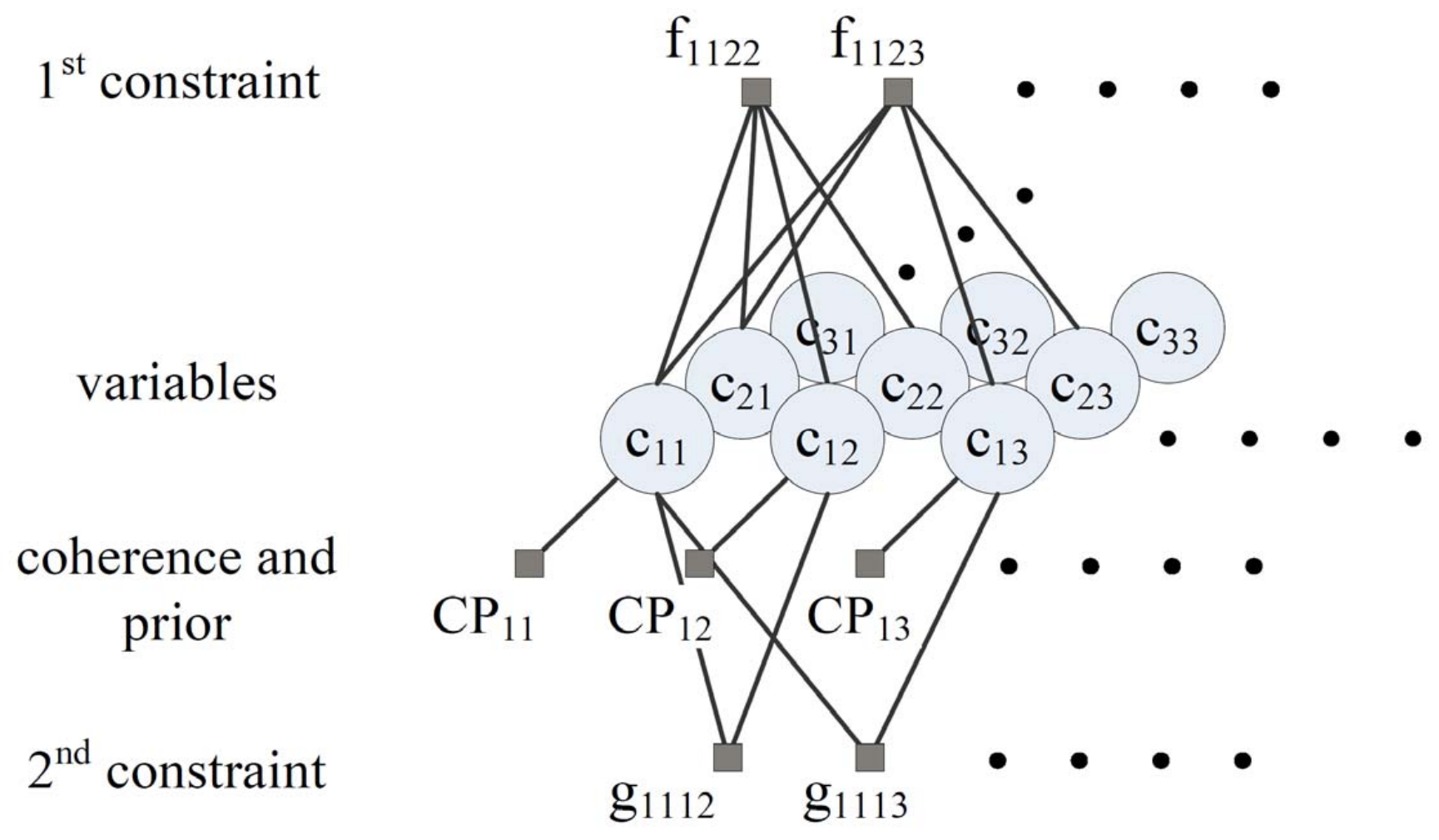
B: the set of biclusters;

w_{ij}: a weight used to cope with missing values;

d: a distance measure based on the model of noise;

a_{ij}^{*}: the ideal value of a_{ij} computed from its exemplar a_{kl}

$$\text{for additive coherence:}\quad a_{ij}^{*}=a_{il}+a_{kj}-a_{kl}$$
$$\text{for multiplicative coherence:}\quad a_{ij}^{*}=a_{il}\times a_{kj}\div a_{kl}$$



The factor graph of the objective function. c_{ij}: the index of the exemplar of a_{ij}; CP, f, g: the factors corresponding to the terms in the objective function.

Regularization (Prior)

1. A penalty term that grows in proportion to the number of biclusters, in order to avoid getting too many small or even single-element biclusters
2. A small penalty based on the size of each bicluster, in order to avoid a bicluster from incorporating any row or column consisting of only background elements

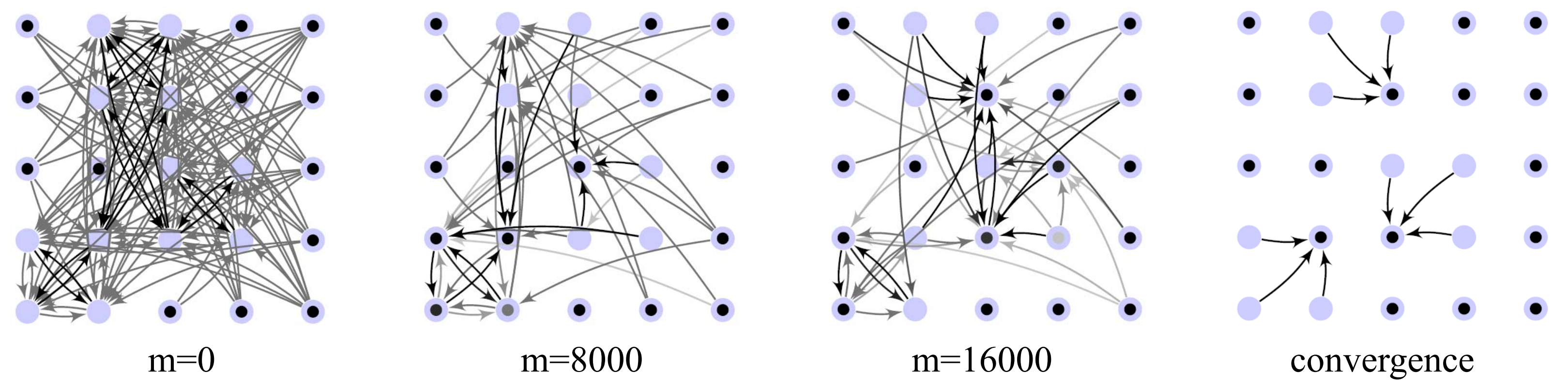
Constraints

1. The set of elements that choose the same exemplar must constitute a valid bicluster, i.e., a submatrix.
2. The exemplar of a bicluster must belong to the bicluster.

3. A Message Passing Algorithm

We derive a message passing algorithm (EBMP), which iteratively passes messages between the factors and variables of the factor graph of our objective function.

- Each message from a factor to a variable is summarized by at most two scalars computed from the messages received by the factor.
- Each message from a variable to a factor is the sum of all the messages received by the variable except the message sent by the factor.
- All messages are initialized to zeroes and then iteratively updated according to an informed schedule until convergence.
- The resulting biclustering is read out from the final message values.



An illustrative example of biclustering using our message passing algorithm. The figure shows the exemplar preferences at different stages of the algorithm (m is the number of messages propagated).

4. A Greedy Algorithm

On very large data matrices encountered in some real-world applications, the message passing algorithm can be too slow to be useful. We introduce a greedy algorithm (EBG) that is significantly faster by settling for a sub-optimal solution.

1. For each element *a* in the data matrix, a candidate bicluster is constructed from the set of elements that prefer *a* as their exemplar
2. Remove duplicates in the resulting biclusters
3. Greedily shrink each remaining bicluster to optimize a modified objective function
4. Do a second round de-duplication and output the biclusters

5. Experiments

5.1 Experiments with Synthetic Data

We create four synthetic datasets containing data matrices of different sizes and noise levels. We compared our two algorithms with Chen&Church [2000] and ROCC [Deodhar, et al., 2009]. Purity (P) and inverse purity (IP) are used to measure the quality of biclustering.

	Chen&Church		ROCC		EBMP		EBG	
	P	IP	P	IP	P	IP	P	IP
10×10 small noise	0.1566	0.4086	0.2263	0.4262	0.9807	0.9722	0.9410	0.9009
10×10 large noise	0.1760	0.5033	0.2877	0.4262	0.8736	0.6445	0.8314	0.6489
50×50 small noise	0.0452	0.4753	0.1339	0.3639	0.7312	0.9664	0.9102	0.7859
50×50 large noise	0.0281	0.0455	0.1164	0.3848	0.7876	0.5839	0.7786	0.6786

The experimental results on synthetic data (each measure is averaged over five runs on ten data matrices.).

5.2 Experiments with Natural Language Bigram Data

In a *word bigram matrix* of a natural language corpus, each row or column is indexed by a word and each element is the frequency of the word bigram in the corpus. Coherent biclusters of a word bigram matrix are useful in inferring grammar rules [Adriaans, et al., 2000; Tu&Honavar, 2008]. A word bigram matrix is usually very sparse, leading to severe background interference in biclustering.

We used a word bigram matrix constructed from the ATIS corpus of the Penn Treebank (matrix size: 334×341; 98.9% of the elements are 0s). We evaluated the biclustering results by comparing the word clusters derived from the biclustering against the word clusters based on the part-of-speech.

	F-measure
Chen&Church	0.2997 (0.0041)
ROCC	0.2909 (0.0131)
EBG	0.4446 (0)

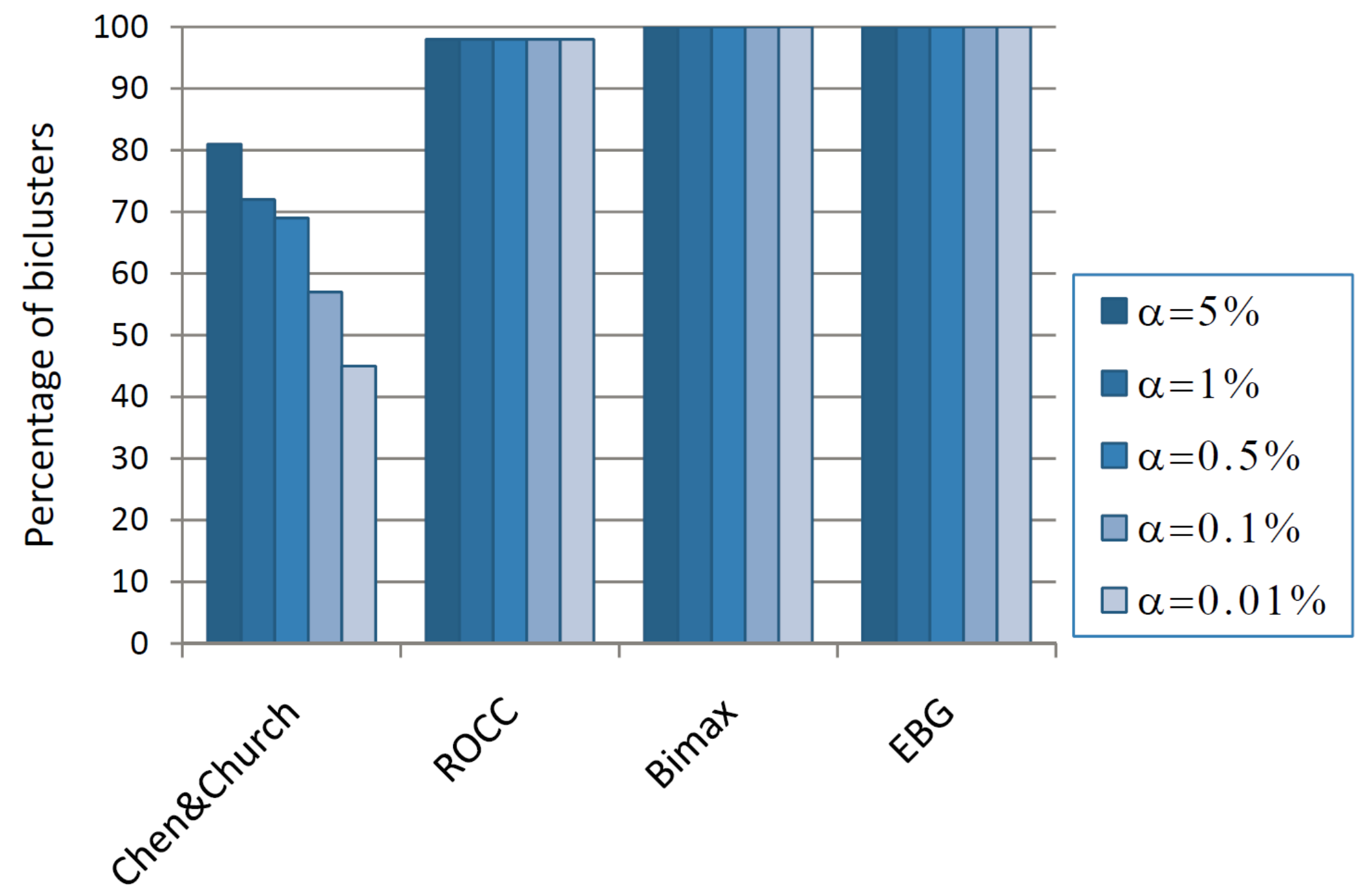
The F-measures averaged from five runs, with the standard deviations in the parentheses.

The table on the right shows some examples of the biclusters extracted by our algorithm. The words that correspond to the bicluster exemplars are italicized.

Row(s)	Column(s)
around, after	eleven, six, *noon*, seven
on, *next*	wednesday, *sunday*, saturday
discount, *business*, coach, first	*class*
coach, expensive, that, lowest, *cheapest*, on	flights, *fare*, fares
is, *serve*, have, are, serves, the	breakfast, *dinner*, lunch
in	phoenix, los, chicago, dallas, toronto, indianapolis, a, minneapolis, westchester, denver, las, washington, *new*, nashville, san, miami, milwaukee

5.3 Experiments with Gene Expression Data

We used the yeast gene expression data matrix from [Gasch, et al., 2000] and followed the experimental setup of [Prelic, et al., 2006]. We compared our algorithm with Chen&Church, ROCC and Bimax [Prelic, et al., 2006].



The experimental results on the yeast gene expression data. The y-axis is the percentage of gene clusters that are overrepresented by at least one Gene Ontology annotation, with five different significance levels (α).

6. Conclusion and Discussion

Our algorithms are competitive with the current state-of-the-art algorithms for finding coherent biclusters. Some interesting directions for future work: improving the scalability of message passing by novel schedules and parallelization, combining the greedy algorithm with the message passing algorithm, etc.